



A New Approach to ESB Migration

Building a foundation for Digital Business

Summary

Traditional business models are being threatened by the digital economy. Having the ability to do business digitally is now a must in order to survive in the long term, let alone thrive. In looking for a way to a digital future, organizations find themselves hamstrung by legacy architectures and products that do not enable their transformation strategy. In many organizations, legacy Enterprise Service Bus implementations have become an obstacle to digital transformation. This paper presents the key drivers of change and discusses an innovative approach to overcoming the legacy ESB hurdle for true digital transformation.

Drivers

Some of the choices made by ESB/SOA technology customers over the years have led them to a dead-end, as their current Enterprise Integration infrastructure does not provide a suitable foundation on which to build for the future. There are many reasons for companies to want to reconsider how integration capabilities are delivered, including:

- Mergers & Acquisition driving a new integration strategy
- Vendor consolidation
- Enterprise-wide integration initiatives
- Technology initiatives requiring new capabilities (e.g., IoT, Analytics, Cloud)
- Vendors introducing next-generation ESB versions lacking full backward compatibility
- The rise of Open Source and Software-as-a-Service
- Concerns or risks with the incumbent supplier and its commercial and licensing practices
- New pricing models providing a better TCO

Commonly used strategies

IT Executives have few options to choose from when faced with inadequate legacy ESB technology:

- **Adopt a new platform for future projects** – The company typically does a vendor evaluation and selects a new platform on which future solutions will be built. The existing integration portfolio is treated as legacy, and the organization must continue to expend resources to support it internally or may decide to outsource the ongoing support of the legacy platform. Projects to migrate all legacy integrations to the new platform often take years to complete.

Highlights of Model-Driven ESB Migration

- MDD enables an automated approach to ESB migration:
 - Cuts time to migrate
 - Results in up to 50% cost reductions
 - Eliminates the risk for human error
 - Preserves 100% of the business logic implemented in the legacy ESB
- The technology is proven and 100% reliable for mission-critical systems
- Approach was used successfully in large migration projects in Europe and North-America: error-free code, on-time delivery and successful go-live

-
- **Rip and Replace** – The company can adopt a new ESB platform and launch a full-scale migration project to decommission the legacy ESB. This must be done while trying to keep up with the constant inflow of requests from the business for new capabilities. Justifying the replacement of existing infrastructure at the expense of meeting new business needs is a challenge for most IT executives.
 - **Status Quo** – The company accepts the solution roadmap proposed by the incumbent vendor including new capabilities to be added in the future, accepting the risk of delayed introduction of the new features. For time-sensitive needs, the customer must supplement the vendor’s stack with commercial add-ons, custom developments, or both.

A new Approach to ESB Migration

A complete replacement of the legacy ESB (“Rip and Replace”) would be the preferred option for many executives if it were not for the associated risk and cost. Bringing in new technology, migrating the entire workload to it within months and decommissioning the legacy sounds ideal.

This is where Model-Driven Development (MDD) comes into play. Although MDD has been around long enough to prove itself time and time again, it is still an underutilized technology.

When comparing ESB’s, it becomes evident that they all incorporate a core set of capabilities that are common. Conceptually this means that one vendor’s ESB can be mapped functionally to another vendor’s ESB. The fact that functional integrity can be preserved makes MDD well-suited for migrating from old to new. The idea is to create a model of each and automate the code generation to migrate the functionality. Of course, it is not a simple undertaking, but it is being done and a description of the approach in more details follows.

Applying MDD to ESB Migrations

Through the use of an MDD-based tool to generate the replacement solution on the target platform, the migration can be accomplished in four major steps (see figure on next page):

- The legacy ESB source code and metadata are introspected to capture the technical and functional requirements. The tool can be built to interpret this information and create a model of the existing implementation. This model is specific to the source (legacy) ESB.
 - The source model is mapped to a pivot (or “Canonical”) model in order to provide a generic representation of ESB capabilities. This is an important step for creating a target solution that includes all of the required functionality because a direct mapping of functionality from source to target would be difficult to achieve.
 - The Canonical model is then mapped to the target ESB model, providing a complete functional representation of the target model. In addition, manual code can be added into the model if desired. For example, this could be done to support unique requirement not provided by the target ESB. The plug-in approach ensures that the solution can be easily regenerated and the manual code automatically added back to the model.
 - Once the target model is complete, the code is generated to run on the target ESB. The generated code runs natively on the target platform without assistance of the migration tool, and it is fully supportable within the target development environment. Furthermore, the code is version-specific to ensure 100% compatibility with the client’s new ESB environment.
-

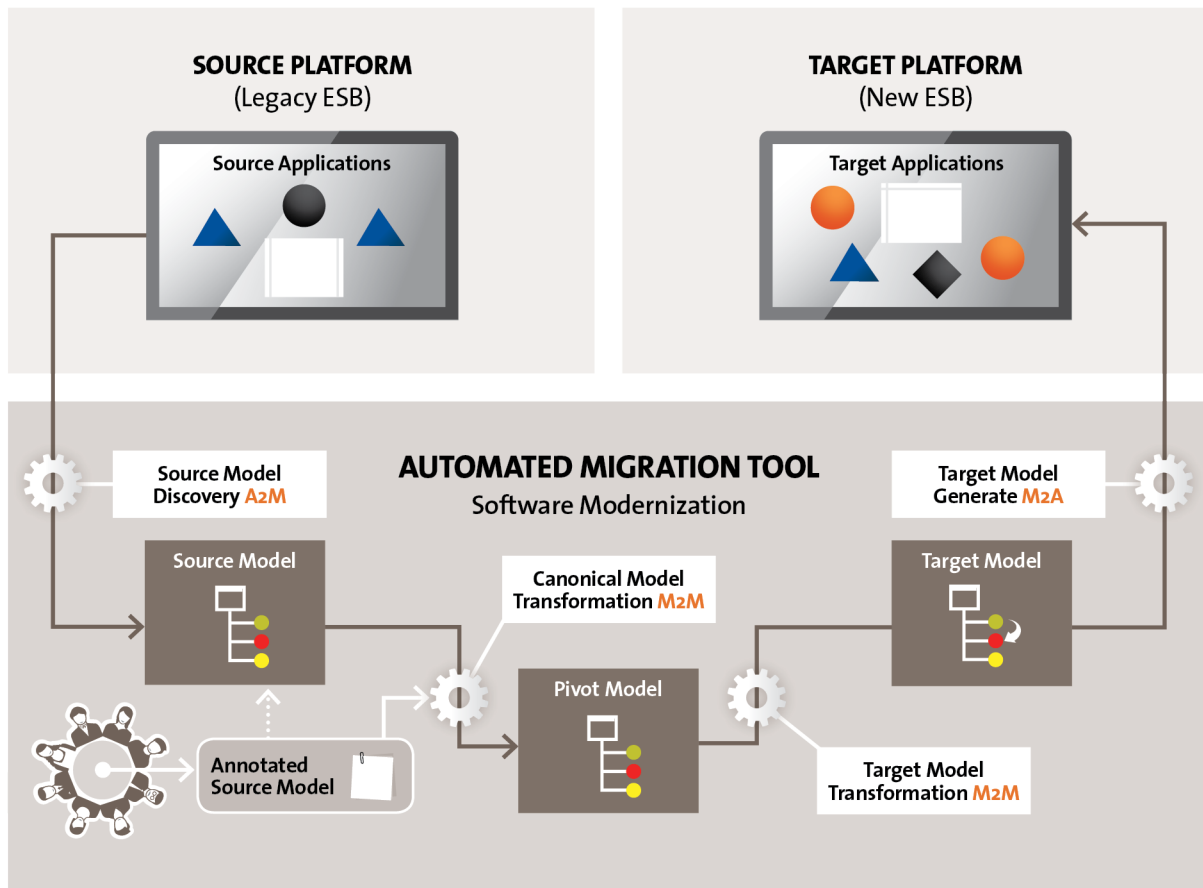


Figure: Mapping the legacy ESB to the Target Architecture

It is important to point out that MDD provides the flexibility to map the legacy ESB to multiple components within an overall target architecture. More options exist than the straight replacement of one ESB with another. This is of particular interest with cloud-native architectures now being broadly adopted and, in many cases, the legacy architecture cannot be simply replicated. Instead, it must be mapped to the new stack, which may be made up of several components such as (1) on premise ESB - open-source or proprietary, (2) Cloud-based ESB, (3) Micro-service architecture, (4) API Management.

The degree of flexibility available to companies wishing to migrate can be summarized as follow:

Basic software migration: The old system is reproduced as-is on the new platform, except for some minor changes that are made to take advantage of unique features available on the target platform.

Software modernization: This is a customized migration, where the company wants to change the way things will be implemented on the target platform (e.g., from file-based integration on the legacy system (mainframe) to Message Queue-based integration on the new platform).

Practical use of the concept

Logimethods uses a MDD-based delivery approach to basically make the “Rip and Replace” option the best option from both a cost and risk standpoint. The MDD-based tool makes the migration predictable, cost-effective and ensures a complete (100%) refactoring of all functionality imbedded in the legacy ESB code. The tool doesn’t rely on documentation that is often incomplete, inaccurate or simply missing. Rather, it extracts all functional requirements from the legacy ESB

meta-data and source code and creates a functionally equivalent solution that runs natively on the target ESB platform. The generated ESB code can be maintained and subsequently modified as any other manually-developed code. A rigorous testing process ensures a 100% match between the old and the new. Furthermore, there is no dependency on the migration tool once the code has been generated.

Benefits of MDD

The following benefits and advantages have been confirmed in the context of complex migration projects:

- A significant acceleration of migration projects. This is made possible by the high degree of automation achieved (typically above 90%) when extracting the requirements from the existing code and generating the new code.
- A functionally equivalent solution is generated to run on the target ESB. A rigorous testing process ensures a 100% match between the old and the new.
- The generated ESB code runs natively on the target platform and can be maintained and subsequently modified as any other manually-developed code. There is no dependency on the migration tool once the code has been generated.
- The generated code adheres to the desired standards and is free of inconsistencies that would otherwise be introduced by a team of off-shore or on-shore developers.
- The new code can be easily regenerated at any point in the migration project if a change must to be made across all or a portion of the deliverables. The risk of introducing manual errors is eliminated even if a widespread change must be applied to previously generated code.
- The knowledge and experience gained from migration projects is added to the migration tool on an ongoing basis. The tool gets better as the project advances.
- As an extra benefit, the resulting solution will likely be better documented and easier to maintain than the source solution.

Case Studies

Large Canadian Bank, Toronto, Canada

This Bank was using a legacy ESB that was inflexible, expensive to operate and inadequately supported by the vendor. The Bank needed to re-platform a large number of interfaces on a modern ESB that would support their current and future needs. Logimethods was brought in to conduct a brief feasibility study for an automated migration to the new platform. We were able to quickly confirm the technical feasibility, provided a fixed-price cost estimate for the entire migration and highlighted the substantial benefits to the Bank of using the proposed approach. Logimethods' Migration Factory accomplished the migration and code testing on time and on budget, while fully meeting all of the technical and business requirements.

National Postal Service, European Country

This national postal service undertook to relocate its entire IT operations to a new facility. In the process, some legacy technologies had to be decommissioned and replaced, including a legacy ESB via which all of the mission-critical mail operation data exchanges between the client and its business partners were conducted. The two most important migration risks were: (1) a lack of documentation and knowledge of the legacy ESB, and (2) the need to perform the transition to a new platform without impacting the mail delivery operations. What was needed was the ability to fully capture all requirements from the source code and create a new solution that was 100% identical functionally. Using its

automated ESB migration tool, Logimethods was able to deliver the new mission critical interfaces on time and on budget.

Closing remarks

Enterprise Integration is a key enabler of the digital enterprise and, for many organizations, the application integration infrastructure built over the years does not fully enable the required business transformation. New capabilities are needed and new architectures must be supported in order to build the “future state”. In the meantime, legacy technology remains critical to run the “current state”. This will result in the duplication of costs and complexity. If used strategically, Model-Driven Development can make the outright replacement and decommissioning of the legacy ESB/EAI the best option from both a cost and risk standpoint. By doing so, organizations can quickly evolve to a unified, seamless and modern enterprise integration architecture fully supporting the digital enterprise.

About Logimethods

Drawing on its strengths and achievements in the field of enterprise integration over the last 15 years, Logimethods is the partner of choice for organizations striving to become digital enterprises. Logimethods provides strategy, delivery and support capabilities to enable a governed and efficient transition to the digital paradigm. Using accelerators, Logimethods delivers on its commitment to provide superior quality and value while speeding up time-to-market. Headquartered in Montreal, with offices in Toronto, Calgary and Chicago, Logimethods also operates a highly efficient solution factory providing on-demand development and managed services, supporting legacy and cloud-native architectures. Logimethods' factory services enable us to serve clients in North-America and globally.

[Contact us](#) to learn more on our ability to help you deliver faster!

